SPECIFICATION

Please amend the specification as follows.

Replace the title with —SYSTEM ON A CHIP HAVING A SYSTEM BUS, AN EXTERNAL BUS, AND A BUS ARBITER WITH PROGRAMMABLE PRIORITIES FOR BOTH BUSES, SOFTWARE, AND METHOD FOR ASSIGNING PROGRAMMABLE PRIORITIES—.

Replace the paragraph beginning on page 5, line 6 with the following:

A1

These assumptions are first that the hybrid scheme of Fig. 2B is used. Second, that the DRAM refresh controller does not make an operation request in steps 5, 6, and 7, and that the CPU ~~422~~ 222 makes an operation request in the step 1, while the remaining functional blocks make continuous operation requests.

Replace the paragraph beginning on page 6, line 1 with the following:

A2

System bus arbiter 394 is for arbitrating requests regarding the system bus 210, and external bus arbiter 396 is for arbitrating requests regarding the external bus 315. The two arbiters 394, 396 perform what is known as hierarchical arbitration relative to each other. System bus arbiter 394 is classified as being in a higher hierarchy than external bus ~~master~~ arbiter 396. In other words, the operation of system bus arbiter 394 is independent, while the operation of external bus arbiter 396 is a subordinate of system bus 394. (And external memory controller 244 serves as a hierarchical bus bridge.) This is illustrated below.

Replace the paragraph beginning on page 7, line 4 with the following:

A3

In other instances, operations are inefficient. For example, in step 2, although the system bus arbiter 394 selects DDMA block #0, the external bus arbiter 396 attempts to select the GDMA channel #2. In this case the GDMA channel #2 actually takes control of the external bus 315, since the functional block selected by the external bus arbiter 396 has a priority over that selected by the system bus arbiter 394. In this case, since the DDMA block #0 has ownership of the system bus 210, while the GDMA channel #2 has ownership of the external bus 315, the DDMA block #0 simply cannot complete its operation in the step 2. As also shown in steps 4 and 8, there are functional blocks that do not complete their operation properly during their allocated step, due to the absence of ownership of the external bus 315. When they do not, they have to wait for another step, while the external bus 315 is being used a3

by another functional block that has ownership of the external bus 315 granted by the external by arbiter 396.

Replace the paragraph beginning on page 11, line 9 with the following:

a4

Another aspect of controller 440 is that it receives inputs from the above-described block 430 directly, through a so-called third path 431, as will be described in more detail below. Third path 431 is distinct from the system bus 410 and the external bus 415.

Replace the paragraph beginning on page 15, line 3 with the following:

a5

Optionally and preferably, external bus controller 810 is adapted to receive inputs directly from the arbiter 495. More specifically, it can receive a select signal SEL from system bus slave selector 740. Alternately, it can receive ~~an external bus grant~~ a drive signal ~~EGNT~~ EDR from external bus master selector 750.

Replace the paragraph beginning on page 16, line 25 with the following:

a6

The multiple channels can program each of all four types of data communication within the GDMA block 430. Even during programming of multiple channels, the GDMA block 430 performs a control such that only one channel operates at one time. The exemplary arbitrary IP blocks to be programmed in each of the GDMA multiple channels may include UART ~~442~~ 552, USB 554, interrupt controller 558, etc.

Replace the paragraph beginning on page 17, line 18 with the following:

a7

The type of bus ownership requests may be represented by four two-bit bus request signals REQ[1:0] = {EREQ, SREQ} using a 1-bit system bus request signal SREQ and a 1-bit external bus request signal EREQ. That is, if REQ[1:0] = 2'b00, no request for a bus is made (no request), and if REQ[1:0] = 2'b01, a request for system bus 410 only is made (system bus only request). If REQ[1:0] = ~~2b'10~~ 2'b10, a request for external bus 415 only is made (external bus only request), and if REQ[1:0] = ~~2b'11~~ 2'b11, a request for both buses is made (both bus request).

Replace the paragraph beginning on page 17, line 25 with the following:

a8

Similarly, a bus ownership grant may be represented by a bus ownership grant signal GNT[1:0] = {EGNT, SGNT}. That is, if GNT[1:0] = 2'b00, no bus is granted (no grant), and if GNT[1:0] = 2'b01, the system bus 410 only is granted (system bus only grant). If a8

GNT[1:0] = ~~2b'10~~ 2'b10, external bus 415 only is granted (external bus only grant), and if GNT[1:0] = ~~2b'11~~ 2'b11, ownership of both buses concurrently is granted (both bus grant).

Replace the paragraph beginning on page 18, line 20 with the following:

a9

Referring now to Fig. 11, it is further advantageous to show the requests of Fig. 9 in the sets indicated by Fig. 10. First, the set of functional blocks making a system bus request by a bus request signal REQ[1:0] equal to 2'b01 or 2'b11, is defined as set S which is the same as the set of system bus masters. Second, the set of functional blocks making an external bus request by a bus request signal REQ[1:0] equal to 2'b10 or 2'b11 is defined as set E. Third, the set of functional blocks making only a system bus request through a bus request signal REQ[1:0] equal to 2'b01 is defined as set SO. Fourth, the set of functional blocks making only an external bus request through a bus request signal REQ[1:0] equal to 2'b10 is defined as set EO which is the same as the set of external bus masters. Fifth, the set of functional blocks making a request for both system bus and external bus through a bus request signal REQ[1:0] equal to 2'b11 is defined as set ES. Sixth, the set of functional blocks making a request for a system bus or an external bus through a bus request signal REQ[1:0] equal to ~~2b'01~~ 2'b01, 2'b10, 2'b11 is defined as set A.